ریاضی و جامعه
شاپا (چاپی): ۶۴۹۳-۲۳۴۵، شاپا(الکترونیکی): ۶۵۰۷-۲۳۴۵
جلد ۴، شماره ۴ (۱۳۹۸)، صص. ۱۴-۱

http://math-sci.ui.ac.ir
http://www.ui.ac.ir

کاربردهایی از آتوماتون متناهی

سمیه تاری

چکیده. مباحث مربوط به زبان‌های صوری و مدل‌های مناسب برای آن‌ها، از مفاهیم پایه و اساسی در رشته علوم کامپیوتر است. از ساده‌ترین مدل‌های محاسبه، آتوماتای متناهی هستند. دلیل سادگی این نوع ماشین‌ها این است که حافظه کمکی در آن‌ها وجود ندارد؛ با وجود این، بسیاری از مسائل را می‌توان با استفاده از آتوماتای متناهی حل کرد. در این مقاله مفهوم آتوماتون متناهی قطعی و غیرقطعی معرفی می‌شود. سپس از آتوماتای متناهی در جستجوی متن برای یک نمونه خاص استفاده می‌شود. هم‌چنین روند خرید اینترنتی با استفاده از آن مدل‌سازی می‌شود.

۱. مقدمه

در سیستم‌های ریاضی مدل‌سازی محاسبه، از مفهوم تخصصی زبان استفاده می‌شود. به هر مجموعه متناهی از نمادها یک الفبا[۱] گفته می‌شود. فرض کنید Σ یک الفبا باشد. هر دنباله متناهی از نمادهای موجود در Σ یک رشته[۲] نامیده می‌شود. رشته پوچ[۳] که با نماد λ نشان داده می‌شود، رشته‌ای است که در آن هیچ نمادی از Σ به کار نرفته است. مجموعه تمام رشته‌های متشکل از اعضای Σ با نماد Σ^* نشان داده می‌شود. هر زیر مجموعه از Σ^* یک زبان[۴] نامیده می‌شود.

یک ماشین (آتوماتون) یک مدل انتزاعی از کامپیوتر است که یک واحد کنترل دارد. واحد کنترل، حالت‌های داخلی دارد که در هر لحظه از زمان می‌تواند در یکی از حالت خود باشد. هر ماشین، یک رشته از یک الفبای مفروض را به عنوان ورودی دریافت می‌کند. ورودی را برحسب واحد کنترل خود می‌خواند، و در پایان یک خروجی ایجاد می‌کند.

در دهه‌ی ۱۹۳۰ تورینگ[۵] ماشینی را تحلیل و بررسی کرد که توانایی‌های کامپیوترهای امروزی را داشت (امروزه مدل مورد بررسی به ماشین تورینگ معروف است). در واقع هدف او این بود که بین کارهایی که ماشین قادر به انجام آن‌ها بود و کارهایی که ماشین قادر به انجام آن‌ها نبود، یک مرزبندی را ایجاد کند. در دهه‌های ۱۹۴۰ و ۱۹۵۰ انواع ساده‌تری از مدل‌های محاسبه که امروزه به آتوماتای متناهی معروف شده‌اند، توسط تیمی متشکل از ریاضیدانان، زیست‌شناسان، روانشناسان، مهندسان و متخصصین اولیه نظریه علوم کامپیوتر بررسی شد. هدف اصلی این پژوهشگران شبیه سازی مغز انسان بوده است [۴].

عبارات و کلمات کلیدی. زبان صوری، آتوماتون متناهی غیرقطعی، آتوماتون متناهی قطعی، جستجوی متن.
نوع مقاله: پژوهشی
دبیر تخصصی: محمدرضا پوریای‌ولی
تاریخ دریافت: ۱۳۹۸/۱۰/۰۶ تاریخ پذیرش: ۱۳۹۹/۰۴/۲۳
http://dx.doi.org/10.22108/msci.2020.120773.1354

[1] Alphabet [2] String [3] Null String [4] Language [5] Turing

در نظریه علوم کامپیوتر، آتوماتای متناهی، از ساده‌ترین مدل‌های محاسبه هستند. آتوماتون متناهی، ماشینی است که تعداد متناهی حالت در واحد کنترل خود دارد و خروجی «بله» یا «نه» را تولید می‌کند. بسیاری از ابزارهای مورد استفاده را می‌توان یک آتوماتون متناهی در نظر گرفت. به عنوان مثال فرآیند روشن یا خاموش کردن یک چراغ با یک آتوماتون متناهی مدل‌سازی می‌شود.

رابین[۶] و اسکات[۷] [۹]، آتوماتای متناهی را معرفی کرده‌اند. آتوماتای متناهی به دو دسته آتوماتای متناهی قطعی و غیرقطعی تقسیم می‌شوند. در آتوماتون متناهی قطعی در هر لحظه از زمان، حرکت ماشین برحسب ورودی و حالت داخلی آن مشخص است؛ در حالی‌که در آتوماتون متناهی غیرقطعی در هر لحظه از زمان، حرکت ماشین برحسب ورودی و حالت داخلی آن وابسته به انتخاب‌های مختلف واحد کنترل است. برای حل برخی از مسائل، استفاده از آتوماتای متناهی قطعی مناسب است و در برخی موارد نیز استفاده از آتوماتای متناهی غیرقطعی مناسب است. رابین و اسکات هم‌ارزی آتوماتای متناهی قطعی و غیرقطعی را اثبات کرده‌اند.

از آتوماتای متناهی در بسیاری از نرم‌افزارها استفاده می‌شود. به عنوان مثال می‌توان به موارد زیر اشاره کرد:

- نرم افزارهای طراحی و کنترل رفتار مدارهای دیجیتال [۵].
- واحد «تحلیل‌گر واژه‌ای» کامپایلر که دریافت متن و تفکیک آن به بخش‌هایی (مثل شناسه‌ها، کلید واژه‌ها و نشانه‌گذاری) را انجام می‌دهد [۴، ۸].
- نرم افزارهای پویش متن‌های بزرگ مانند مجموعه صفحات وب، برای یافتن وجود واژه، عبارت یا یک چیز خاص [۱، ۲].
- نرم افزارهای کنترل سیستم‌هایی که تعدادی متناهی حالت مجزا دارند مانند پروتکل‌های ارتباطی یا پروتکل‌های امن تبادل داده [۳، ۴].

در بخش ۲ از این مقاله، به طور مختصر توصیف ریاضی آتوماتون متناهی قطعی و غیرقطعی بیان می‌شود. در بخش ۳، آتوماتون متناهی غیرقطعی در مدل‌سازی روند خرید اینترنتی استفاده می‌شود. در بخش ۴، کاربرد آتوماتون متناهی در جستجوی کلمات کلیدی بیان می‌شود.

۲. آتوماتون متناهی قطعی و غیرقطعی

در این بخش، مفهوم انتزاعی آتوماتون متناهی قطعی و غیرقطعی معرفی می‌شود و با وجود تفاوت در عملکرد آتوماتون متناهی قطعی و غیر قطعی، نشان داده می‌شود که قدرت محاسباتی هر دو یکسان است [۶، ۷، ۱۰].

اگر Σ یک الفبا باشد. یک آتوماتون متناهی قطعی، به صورت شهودی وسیله محاسبه‌گری (الگوریتمی) است که به ازای هر رشته $x \in \Sigma^*$، به عنوان ورودی، خروجی «بله» یا «نه» را تولید می‌کند، و زبان پذیرفته‌شده برای یک آتوماتون متناهی قطعی، مجموعه تمام رشته‌هایی است که به ازای آن‌ها خروجی بله تولید می‌شود. آتوماتون متناهی قطعی با استفاده از نمادهای ریاضی (بخصوص توابع)، جدول انتقال یا دیاگرام انتقال تعریف می‌شود.

تعریف ۱.۲. *یک آتوماتون متناهی قطعی*[۸] *پنج تایی* $M = (Q, \Sigma, q_0, A, \delta)$ *است که در آن:*

- Q *یک مجموعه متناهی است که آن‌را مجموعه حالت‌های* M *می‌نامند.*
- Σ *یک الفبا است.*
- $q_0 \in Q$ *حالت شروع است. محاسبه در حالت* q_0 *آغاز می‌شود.*
- A *یک زیرمجموعه از* Q *است که آن‌را مجموعه حالت‌های پایانی می‌نامند. در حالتی که محاسبه در حالت پایانی تمام شود به منزله رسیدن به جواب «بله» است. در غیر این‌صورت به منزله رسیدن به جواب «نه» است.*

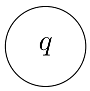
[6] Rabin [7] Scott [8] Deterministic Finite Automaton

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354

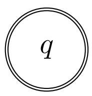
- δ یک تابع از $Q \times \Sigma$ به Q است که آن‌را تابع انتقال[۹] می‌نامند. تابع δ نحوه عملکرد M را توصیف می‌کند.

تبصره ۲.۲. *تابع انتقال $\delta : Q \times \Sigma \to Q$، با استفاده از دیاگرام انتقال با ویژگی‌های زیر مشخص می‌شود:*

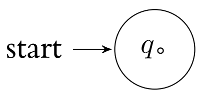
- هر حالت $q \in Q \setminus A \cup \{q_0\}$ به صورت زیر در دیاگرام نشان داده می‌شود:



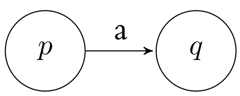
- هر حالت $q \in A$ به صورت زیر در دیاگرام نشان داده می‌شود:



- حالت شروع به صورت زیر در دیاگرام نشان داده می‌شود:



- $\delta(p, a) = q$ به صورت زیر در دیاگرام نشان داده می‌شود:

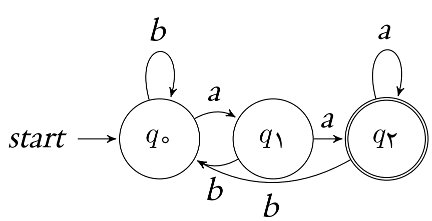


مثال ۳.۲. *آتوماتون متناهی $M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \{q_2\}, \delta)$ را در نظر بگیرید که در آن تابع δ در جدول ۱ آورده شده است.*

جدول ۱: جدول انتقال M مربوط به مثال ۳.۲

δ	a	b
q_0	q_1	q_0
q_1	q_2	q_0
q_2	q_2	q_0

به عبارت دیگر $\delta(q_0, b) = \delta(q_1, b) = \delta(q_2, b) = q_0$، $\delta(q_0, a) = q_1$، $\delta(q_1, a) = q_2$ و $\delta(q_2, a) = q_2$. دیاگرام انتقال M در شکل ۱ آورده شده است.



شکل ۱: دیاگرام انتقال مربوط به مثال ۳.۲

[9] Transition Function

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354

هر آتوماتون متناهی $M=(Q,\Sigma,q_{\circ},A,\delta)$ به ازای هر رشته ورودی $x\in\Sigma^*$، در حالت شروع $q_{\circ}$ با خواندن اولین نماد x محاسبه خود را شروع می‌کند. برحسب تابع انتقال خود، تغییر حالت می‌دهد و نمادهای بعدی را یکی پس از دیگری می‌خواند. در نهایت به انتهای رشته می‌رسد که محاسبه را در این مرحله با حالت پایانی یا حالت غیرپایانی برای رشته x تمام می‌کند. رشته $x=baa$ را در مثال ۳.۲ در نظر بگیرید. هنگام شروع محاسبه، M در حالت $q_{\circ}$ نماد b را خوانده، به نماد بعدی می‌رسد. طبق تابع انتقال، حالت آن $q_{\circ}$ می‌شود. اکنون در حالت $q_{\circ}$ نماد a را خوانده به نماد بعدی می‌رسد. طبق تابع انتقال خود، حالت آن q_1 می‌شود. در این مرحله نماد a را در حالت q_1 خوانده و به انتهای رشته می‌رسد. طبق تابع انتقال، حالت آن q_2 می‌شود. بنابراین در انتهای محاسبه به حالت پایانی q_2 می‌رسد. به همین ترتیب برای رشته $x=aab$ در انتهای محاسبه به حالت غیرپایانی $q_{\circ}$ می‌رسد.

برای به دست آوردن توصیف دقیق مفهوم شهودی محاسبه انجام شده توسط آتوماتون متناهی قطعی (به عنوان مثال روند توصیف شده در بالا برای رشته $x=baa$)، توسیع تابع انتقال $\delta:Q\times\Sigma\to Q$ به تابع $\delta^*:Q\times\Sigma^*\to Q$ لازم است که با استفاده از استقرا انجام می‌شود.

تعریف ۴.۲. *فرض کنید $M=(Q,\Sigma,q_{\circ},A,\delta)$ یک آتوماتون متناهی قطعی باشد. دراین‌صورت تابع انتقال توسعه یافته*[10] *$\delta^*:Q\times\Sigma^*\to Q$ به صورت استقرایی زیر تعریف می‌شود:*

- *به ازای هر $q\in Q$، $\delta^*(q,\lambda)=q$.*
- *$\delta^*(q,y\sigma)=\delta(\delta^*(q,y),\sigma)$ به ازای هر $q\in Q$، $y\in\Sigma^*$ و $\sigma\in\Sigma$.*

مثال ۵.۲. *آتوماتون متناهی قطعی مثال ۳.۲ را در نظربگیرید. در این‌صورت*

$$\delta^*(q_{\circ},aaba)=\delta(\delta^*(q_{\circ},aab),a)=\delta(\delta(\delta^*(q_{\circ},aa),b),a)=\delta(\delta(\delta(\delta^*(q_{\circ},a),a),b),a)=$$
$$\delta(\delta(\delta(\delta(q_{\circ},a),a),b),a)=q_1$$

با استفاده از تابع انتقال توسعه یافته، تعریف ریاضی زبان پذیرفته‌شده برای آتوماتون متناهی قطعی بیان می‌شود.

تعریف ۶.۲. *الف) فرض کنید $M=(Q,\Sigma,q_{\circ},A,\delta)$ یک آتوماتون متناهی قطعی باشد. زبان پذیرفته‌شده برای M که با نماد $L(M)$ نشان داده می‌شود، عبارت است از:*

$$L(M)=\{x\in\Sigma^*:\quad \delta^*(q_{\circ},x)\in A\}$$

ب) زبان $L\subseteq\Sigma^*$ محاسبه‌پذیر توسط یک آتوماتون متناهی قطعی است هرگاه آتوماتون متناهی قطعی مانند $M=(Q,\Sigma,q_{\circ},A,\delta)$ موجود باشد که $L=L(M)$.

یک آتوماتون متناهی غیرقطعی، از بسیاری جهات شبیه آتوماتون متناهی قطعی است. تفاوت عمده آن‌ها در تعریف تابع انتقال است. در هر آتوماتون متناهی قطعی پردازش هر رشته برحسب تابع انتقال به صورتی یکتا انجام می‌شود؛ اما در آتوماتای متناهی غیرقطعی به دلیل تعریف تابع انتقال به صورت خاص، امکان پردازش یک رشته به روش‌های مختلف وجود دارد.

تعریف ۷.۲. *یک آتوماتون متناهی غیرقطعی*[11] *پنج تایی $M=(Q,\Sigma,q_{\circ},A,\delta)$ است که در آن:*

- *Q یک مجموعه متناهی است که آن‌را مجموعه حالت‌های M می‌نامند.*
- *Σ یک الفبا است.*
- *$q_{\circ}\in Q$ حالت شروع است.*

[10] Extended Transition Function [11] Nondeterministic Finite Automaton

ـ A یک زیرمجموعه از Q است که آن‌را مجموعه حالت‌های پایانی می‌نامند.

ـ $\delta : Q \times \Sigma \cup \{\lambda\} \to \mathcal{P}(Q)$ تابع انتقال M است که در آن $\mathcal{P}(Q)$ مجموعه تمام زیرمجموعه‌های Q است.

تبصره ۸.۲. *توضیح نکات زیر در مورد تابع انتقال آتوماتون متناهی غیرقطعی $M = (Q, \Sigma, q_0, A, \delta)$ ضروری است:*

- اگر $\delta(q, \sigma) = \emptyset$، آنگاه M در حالت q با خواندن نماد σ، هیچ تبدیل حالت نخواهد داشت و محاسبه متوقف می‌شود.
- اگر $\delta(q, \sigma) = \{p_1, \ldots, p_n\}$، آنگاه M در حالت q با خواندن نماد σ، امکان تبدیل حالت به حالت‌های p_1،...،p_n (به عبارت دیگر n انتخاب) را دارد.
- چون λ رشته پوچ است، لذا $\delta(q, \lambda) = \{p_1, \ldots, p_n\}$ به معنای این است که M در حالت q امکان تغییر حالت به حالت‌های p_1،...و p_n را بدون خواندن نمادی از رشته ورودی دارد. در اصطلاح M در حالت q، روی رشته ورودی در جا می‌زند؛ ولی حالت آن می‌تواند به یکی از حالت‌های p_1،...،p_n تغییر کند.

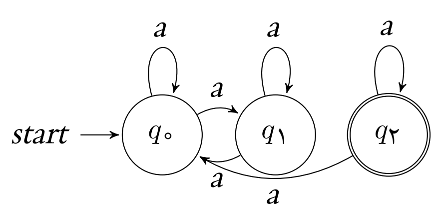
برای رسم دیاگرام انتقال مربوط به آتوماتای متناهی غیرقطعی از نمادهای معرفی شده برای رسم دیاگرام انتقال آتوماتای متناهی قطعی استفاده می‌شود.

مثال ۹.۲. *آتوماتون متناهی غیرقطعی $M = (\{q_0, q_1, q_2\}, \{a, b\}, q_0, \{q_2\}, \delta)$ را در نظر بگیرید که جدول ۲ تابع انتقال آن را توصیف می‌کند.*

جدول ۲: جدول انتقال M مربوط به مثال ۹.۲

δ	a	b	λ
q_0	$\{q_0, q_1\}$	$\emptyset$	$\emptyset$
q_1	$\{q_0, q_1\}$	$\emptyset$	$\emptyset$
q_2	$\{q_0, q_2\}$	$\emptyset$	$\emptyset$

به عبارت دیگر، $\delta(q_2, a) = \{q_2, q_0\}$، $\delta(q_1, a) = \{q_0, q_1\}$، $\delta(q_0, a) = \{q_0, q_1\}$ و بقیه موارد متناظر با مجموعه تهی است. دیاگرام متناظر با آن در شکل ۲ آورده شده است.



شکل ۲: دیاگرام انتقال مربوط به مثال ۹.۲

فرض کنید $M = (Q, \Sigma, q_0, A, \delta)$ یک آتوماتون متناهی غیرقطعی باشد. در این‌صورت منظور از زبان پذیرفته شده برای M مجموعه xهای متعلق به Σ^* است که M در حالت شروع با خواندن نمادهای x از سمت چپ یکی پس از دیگری، بر حسب تابع انتقال δ، حداقل از یکی از مسیرهای محاسبه به حالت پذیرش برسد. برای توصیف ریاضی زبان

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354

پذیرفته شده برای M، $L(M)$، توسیع تابع انتقال $\delta : Q \times \Sigma \cup \{\lambda\} \to \mathcal{P}(Q)$ به تابع $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ لازم است. با توجه به اینکه این امکان وجود دارد که برای $n \in \mathbb{N}$، $\delta^n(p, \lambda)$ یک مجموعه‌ای با بیش از یک عضو باشد، بنابراین برای تعریف تابع δ^* تعریف زیر مورد نیاز است.

تعریف ۱۰.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک آتوماتون متناهی غیرقطعی باشد. در این‌صورت برای هر $q \in Q$، $\delta^n(q, \lambda)$ با استفاده از استقرا روی n به صورت زیر تعریف می‌شود:*

$$\delta^1(q, \lambda) = \delta(q, \lambda) \cup \{q\}$$
$$\delta^{n+1}(q, \lambda) = \delta^n(q, \lambda) \cup \{\delta^1(p, \lambda) : \quad p \in \delta^n(q, \lambda)\}$$

تعریف ۱۱.۲. *فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک آتوماتون متناهی غیرقطعی باشد. در این‌صورت تابع انتقال توسعه یافته $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ به صورت استقرایی زیر تعریف می‌شود:*

- *به ازای هر $q \in Q$، قرار دهید:*

$$\delta^*(q, \lambda) = \delta^n(q, \lambda)$$

که در آن $\delta^n(q, \lambda) = \delta^{n+1}(q, \lambda)$.

- *به ازای هر $q \in Q$، هر $y \in \Sigma^*$ و هر $\sigma \in \Sigma$ قرار دهید:*

$$\delta^*(q, y\sigma) = \bigcup_{p \in \delta^*(q,y)} \delta(p, \sigma) \cup \bigcup_{p \in \delta^*(q,y)} \delta^*(\delta(p, \sigma), \lambda)$$

مثال ۱۲.۲. *آتوماتون متناهی غیرقطعی مثال ۹.۲ را در نظر بگیرید. در این‌صورت*

$$\delta^*(q_\circ, \lambda) = \{q_\circ\}$$
$$\delta^*(q_\circ, a) = \bigcup_{p \in \delta^*(q_\circ,\lambda)} \delta(p, a) \cup \bigcup_{p \in \delta^*(q_\circ,\lambda)} \delta^*(\delta(p, a), \lambda) = \delta(q_\circ, a) \cup \delta^*(q_\circ, \lambda) = \{q_\circ, q_1\}$$

تعریف ۱۳.۲. *الف) فرض کنید $M = (Q, \Sigma, q_\circ, A, \delta)$ یک آتوماتون متناهی غیرقطعی باشد. زبان پذیرفته شده برای M که با نماد $L(M)$ نشان داده می‌شود، عبارت است از:*

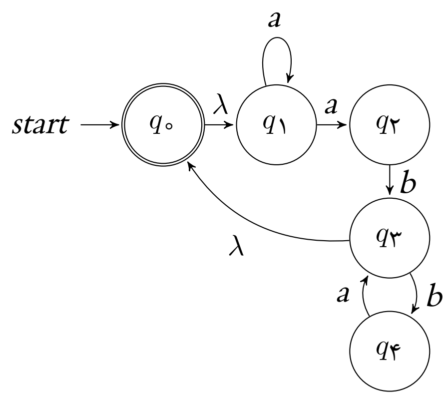
$$L(M) = \{x \in \Sigma^* : \quad \delta^*(q_\circ, x) \cap A \neq \emptyset\}$$

ب) زبان $L \subseteq \Sigma^*$، محاسبه‌پذیر توسط آتوماتون متناهی غیرقطعی گفته می‌شود هرگاه آتوماتون متناهی غیرقطعی مانند $M = (Q, \Sigma, q_\circ, A, \delta)$ موجود باشد که $L = L(M)$.

مجموعه زبان‌های محاسبه‌پذیر توسط آتوماتای متناهی غیرقطعی، همان مجموعه زبان‌های محاسبه‌پذیر توسط آتوماتای متناهی قطعی است. به عبارت دیگر اگر چه در آتوماتای متناهی غیرقطعی مسیرهای محاسباتی مختلف وجود دارد؛ ولی نسبت به آتوماتای متناهی قطعی قدرت محاسباتی بیشتری را ندارد و تنها انجام محاسبات را ساده‌تر می‌کند. به وضوح هر آتوماتون متناهی قطعی یک آتوماتون متناهی غیرقطعی است. در مثال ۱۴.۲ نحوه تبدیل یک آتوماتون متناهی غیرقطعی به آتوماتون متناهی قطعی توصیف می‌شود. هم‌ارزی این دو مفهوم نیز به همین روش اثبات می‌شود.

مثال ۱۴.۲. *فرض کنید دیاگرام انتقال آتوماتون متناهی غیرقطعی $M = (Q, \Sigma, q_\circ, A, \delta)$ به صورت شکل ۳ باشد.*

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354



شکل ۳: دیاگرام انتقال مربوط به مثال ۲.۱۴

تبدیل M به یک آتوماتون متناهی قطعی در دو مرحله زیر انجام می‌شود:

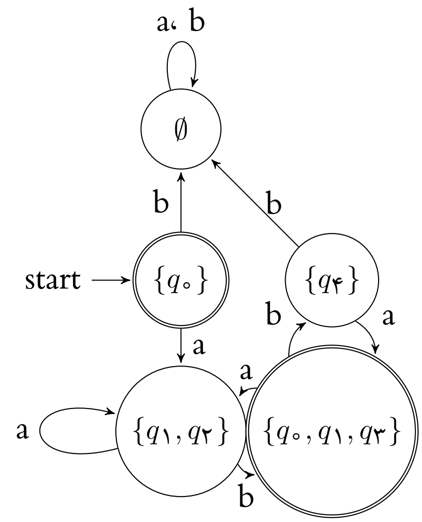
مرحله ۱) ابتدا M تبدیل به آتوماتون متناهی غیرقطعی $M_۱ = (Q, \Sigma, q_۰, A, \delta_۱)$ می‌شود که در آن $\delta_۱(p, \lambda) = \emptyset$ برای هر $p \in Q$. برای این منظور کافی است برای هر $\sigma \in \Sigma$ و $p \in Q$ قرار داده شود $\delta_۱(p, \sigma) = \delta^*(p, \sigma)$. بنابراین با محاسبه δ^* تابع انتقال $\delta_۱$ محاسبه می‌شود.

مرحله ۲) اکنون آتوماتون متناهی قطعی $M_۲ = (Q_۱, \Sigma, \{q_۰\}, A_۱, \delta_۲)$ را چنان در نظر بگیرید که در آن $Q_۱ = \mathcal{P}(Q)$، $A_۱ = \{q \subseteq Q : \ q \cap A \neq \emptyset\}$ و $\delta_۲ : \mathcal{P}(Q) \times \Sigma \to \mathcal{P}(Q)$ تابع انتقالی است که $\delta_۲(q, \sigma) = \bigcup_{p \in q} \delta_۱(p, \sigma)$. واضح است که $L(M) = L(M_۲)$.

چون مجموعه حالات آتوماتون متناهی قطعی $M_۲$ همان مجموعه توانی Q است که یک مجموعه ۵ عضوی است. بنابراین آتوماتون متناهی قطعی $M_۲$، $۳۲ = ۲^۵$ حالت دارد که در عمل بسیاری از این حالت‌ها غیرضروری هستند. به عبارت دیگر، بسیاری از این حالت‌ها متعلق به هیچ $\delta_۲(q, \sigma)$ نیستند. در اصطلاح این حالت‌ها دسترس ناپذیر گفته می‌شوند که حذف آن‌ها از دیاگرام انتقال، هیچ تاثیری در محاسبات آتوماتون مورد نظر ندارد. برای حذف حالت‌های دسترس ناپذیر به صورت زیر عمل می‌شود:

ابتدا برای هر $\sigma \in \Sigma$، مقادیر $\delta_۲(\{q_۰\}, \sigma)$ محاسبه می‌شود و حالت‌های متناظر با آن‌ها به دیاگرام انتقال اضافه می‌شوند. چون $\delta_۲(\{q_۰\}, a) = \delta_۱(q_۰, a) = \{q_۱, q_۲\}$ و $\delta_۲(\{q_۰\}, b) = \delta_۱(q_۰, b) = \emptyset$. لذا حالت‌های $\{q_۱, q_۲\}$ و $\emptyset$ به دیاگرام انتقال اضافه می‌شوند. سپس برای حالت‌های اضافه شده نیز مقادیر متناظر با آن‌ها محاسبه می‌شوند. در صورت نیاز حالت‌های جدید به دیاگرام انتقال اضافه می‌شوند. اضافه کردن حالت‌های جدید تا زمانی ادامه می‌یابد که در یک مرحله حالت جدیدی حاصل نشود. در پایان از حالت‌های موجود در دیاگرام انتقال، حالت‌های پایانی مشخص می‌شوند. در شکل ۴ دیاگرام انتقال $M_۲$ آورده شده است. البته لازم به توضیح است که در برخی موارد به جای دیاگرام انتقال از جدول انتقال نیز استفاده می‌شود.

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354



شکل ۴: دیاگرام انتقال $M_۲$ مربوط به مثال ۱۴.۲

۳. آتوماتون متناهی غیرقطعی و خرید اینترنتی

در این بخش، کارکرد آتوماتای متناهی غیرقطعی در روند خرید اینترنتی مطالعه می‌شود. مدل مطرح شده توسط هاپ‌کرافت[۱۲] و آلمن[۱۳] [۴]، با جزئیات تکمیلی در اینجا توصیف می‌شود. مشتری برای خرید اینترنتی از پول الکترونیکی استفاده می‌کند و فروشنده مطمئن است که پول دریافتی واقعی است و امکان کپی از آن برای استفاده دوباره وجود ندارد. بنابراین فایل پول الکترونیکی باید توسط بانک با استفاده از یک سیستم کدگذاری، کدگذاری شود و یک بانک اطلاعاتی داده‌ای از فایل‌های معتبر صادر شده در بانک‌ها موجود باشد که واقعی بودن پول الکترونیکی را برای فروشنده تضمین نماید[۱۴].

به منظور استفاده از پول الکترونیکی باید روش‌هایی به کار گرفته شود که انعطاف لازم را برای مشتری فراهم کند. همچنین صرف نظر از اینکه مقدار پول چقدر است باید یک سیاست مطمئن پیاده‌سازی شود که فقط اجازه انجام کارهای مجاز را بدهد. به عنوان مثال اجازه دزدیدن از دیگران یا ساختن پول تقلبی را ندهد.

۱.۳. وظایف اساسی. در این مثال سه مشارکت‌کننده وجود دارد: مشتری، فروشگاه و بانک. برای راحتی کار فرض بر این است که فقط یک نوع فایل پول الکترونیکی وجود دارد. مشتری می‌خواهد پول الکترونیکی را به حساب فروشگاه واریز کند، صاحب فروشگاه می‌تواند آنرا از بانک وصول کند و فروشگاه اجناس را به مشتری ارسال می‌کند. تعامل میان این سه مشارکت‌کننده (مشتری، فروشگاه و بانک) به پنج حالت زیر محدود می‌شود:

۱ــ مشتری برای خرید کالا، عمل پرداخت [۱۵] را انجام می‌دهد. عمل پرداخت با نماد P در دیاگرام انتقال نشان داده می‌شود.

۲ــ فروشگاه عمل درخواست وصول [۱۶] پول را همراه با ارسال یک پیغام به بانک انجام می‌دهد. عمل درخواست با نماد R در دیاگرام انتقال نشان داده می‌شود.

---

[۱۴] در اینجا جنبه کدگذاری پول یا نحوه نگهداری فایل‌های معتبر پول الکترونیکی بررسی نمی‌شود.

[12]Hopcroft [13]Ullman [15]Pay [16]Redeem

۳ــ بانک عمل انتقال پول [۱۷] را با ایجاد یک پول الکترونیکی جدید و کدگذاری شده به حساب فروشگاه انجام می‌دهد. عمل انتقال پول با نماد T در دیاگرام انتقال نشان داده می‌شود.

۴ــ فروشگاه عمل ارسال [۱۸] کالا را انجام می‌دهد. عمل ارسال با نماد S در دیاگرام انتقال نشان داده می‌شود.

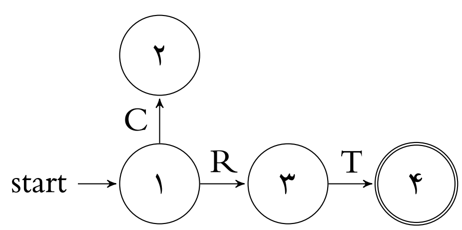
۵ــ مشتری می‌تواند عمل لغو خرید [۱۹] را انجام دهد که با یک پیغام درخواست برگشت پول به حساب مشتری به بانک فرستاده می‌شود. عمل لغو خرید با نماد C در دیاگرام انتقال نشان داده می‌شود.

۲.۳. مدل‌سازی. روند خرید اینترنتی، با استفاده از آتوماتای متناهی غیرقطعی مدل‌سازی می‌شود. هر حالت بیانگر وضعیتی خواهد بود که هر کدام از مشارکت کننده‌ها می‌توانند در آن دخیل باشند. تغییر حالت زمانی اتفاق می‌افتد که یکی از پنج مورد ذکر شده اتفاق بیفتد. اگر چه هر مشارکت کننده آغازگر یک یا چندین رویداد است، ولی این رویدادها عوامل بیرونی برای آتوماتون اصلی در نظر گرفته می‌شود. به عبارت دیگر، رخدادی که می تواند روی دهد مهم است نه مشارکت‌کننده‌ای که آنرا آغاز می‌کند. در ادامه آتوماتون متناهی مربوط به هریک از مشارکت‌کننده‌ها آورده می‌شود. در هرمورد فقط رویدادهایی نشان داده می‌شود که روی مشارکت‌کننده تأثیر می‌گذارد. برای مثال، عمل پرداخت فقط روی مشتری و فروشگاه تأثیر می‌گذارد.

شکل ۵، آتوماتون متناهی مربوط به عملکرد بانک است. حالت ۱، حالت شروع است که نشانگر این است که بانک پول الکترونیکی را صادر کرده فارغ از اینکه درخواستی برای وصول یا لغو تراکنش داشته باشد. به محض اینکه درخواست لغو تراکنش توسط مشتری به بانک فرستاده شود، حالت ۱ به حالت ۲ تبدیل می‌شود و پول توسط بانک به حساب مشتری برگشت داده می‌شود. در واقع حالت ۲ نشان دهنده وضعیتی است که تراکنش لغو شده است. همچنین در حالت ۱، بانک با دریافت درخواست فروشگاه برای وصول پول، به حالت ۳ رفته و تغییر حالت ۳ به حالت ۴ با انتقال پول به حساب فروشگاه انجام می‌شود.

شکل ۶، آتوماتون متناهی مربوط به عملکرد فروشگاه است. حالت a حالت شروع است که با انجام عمل پرداخت توسط مشتری به حالت b تبدیل می‌شود. در حالت b فروشگاه با ارسال پیام درخواست وصول پول به بانک وارد حالت c می‌شود. با دریافت پیام انتقال پول از بانک حالت c به حالت d تبدیل می‌شود. حالت d با ارسال کالا به مشتری وارد حالت نهایی e می‌شود. در این مرحله فرآیند فروش کالا توسط فروشگاه با موفقیت به اتمام می‌رسد.

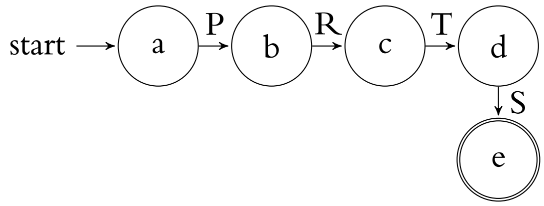
شکل ۷، آتوماتون متناهی مربوط به عملکرد مشتری است. آتوماتون مورد نظر فقط یک حالت دارد که نشان دهنده آزادی عمل مشتری است و مشتری می‌تواند به طور متناوب عمل خرید و لغو خرید انجام دهد.



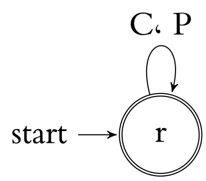
شکل ۵: دیاگرام انتقال بانک

[17] Transfer [18] Ship [19] Cancel

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354



شکل ۶: دیاگرام انتقال فروشگاه



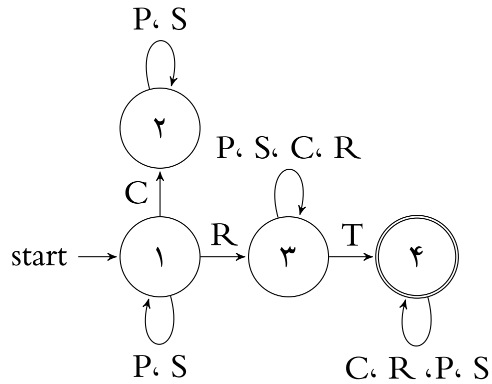
شکل ۷: دیاگرام انتقال مشتری

۳.۳. آتوماتون اصلی خرید اینترنتی. سه آتوماتون توصیف شده، عملکرد سه مشارکت‌کننده (مشتری، فروشگاه و بانک) را به طور مستقل نشان می‌دهند. از محاسبه موازی و هم‌زمان سه آتوماتون، برای به دست آوردن آتوماتون اصلی استفاده می‌شود. در اینجا دو مشکل اساسی وجود دارد:

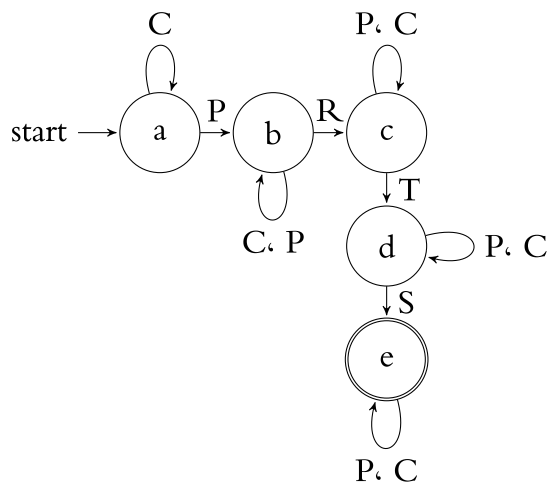
۱) انتقال‌هایی موجودند که برخی از این آتوماتون‌ها، آن‌ها را شامل نمی‌شوند. وقتی سه آتوماتون با هم در یک آتوماتون کلی در نظر گرفته می‌شوند، آنگاه عدم تعریف هر کدام از مشارکت‌کننده‌ها باعث از کار افتادن آتوماتون اصلی می‌شود. به عنوان مثال عمل پرداخت توسط بانک انجام نمی‌شود و در طول عملیات بانک هرگز عمل پرداخت اتفاق نمی‌افتد؛ ولی در آتوماتون اصلی ممکن است ماشین بانک با ورودی P مواجه شود. برای تکمیل دیاگرام انتقال، اثر بانک در هرحالتی برای P خود آن حالت تعریف می‌شود. در این‌صورت آتوماتون مربوط به بانک در شکل ۵ وقتی با P مواجه شود می‌تواند در هر حالتی که هست بماند. بنابراین اثر هر حالت در هر دیاگرام، روی اعمال غیر مرتبط خود آن حالت تعریف می‌شود. با اینکار این مشکل در دیاگرام اصلی حل می‌شود.

۲) مشکل محتمل دیگر این است که یکی از مشارکت‌کننده‌ها به صورت عمدی یا غیر عمدی پیغام غیر منتظره‌ای را بفرستد. به عنوان مثال هنگامی که فروشگاه روی حالت c در شکل ۶ قرار دارد، مشتری عمل پرداخت را برای دومین بار انجام دهد. چون این حالت هیچ کمان خروجی با برچسب P ندارد (به عبارت دیگر $\delta(c, P) = \emptyset$)، لذا ماشین فروشگاه قبل از دریافت پیغام دریافت بانک از کار می‌افتد. برای جلوگیری از کار افتادن سیستم، با انجام کاری مشابه قسمت ۱ این مشکل نیز حل می‌شود.

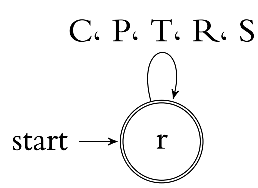
در شکل‌های ۸، ۹ و ۱۰ دیاگرام‌های جدید مربوط به بانک، فروشگاه و مشتری آورده شده است که از آن‌ها در آتوماتون اصلی استفاده می‌شود.



شکل ۸: دیاگرام مورد استفاده در آتوماتون اصلی مربوط به بانک



شکل ۹: دیاگرام مورد استفاده در آتوماتون اصلی مربوط به فروشگاه



شکل ۱۰: دیاگرام مورد استفاده در آتوماتون اصلی مربوط به مشتری

اکنون آتوماتون اصلی $M = (Q, \Sigma, q_\circ, A, \delta)$ را در نظر بگیرید که در آن:

- $Q = \{r\} \times \{۱, ۲, ۳, ۴\} \times \{a, b, c, d, e\}$

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354

- $\Sigma = \{C, P, T, R, S\}$
- $q_۰ = (r, ۱, a)$
- $A = \{(r, ۴, e)\}$
- δ تابعی از $Q \times \Sigma$ به مجموعه $\mathcal{P}(Q)$ با ضابطه $\delta((r, i, y), X) = (r, \delta_۱(i, X), \delta_۲(y, X))$ است که در آن $\delta_۱$ تابع انتقال بانک و $\delta_۲$ تابع انتقال فروشگاه هستند.

۴. آتوماتون متناهی قطعی و جستجوی کلمه

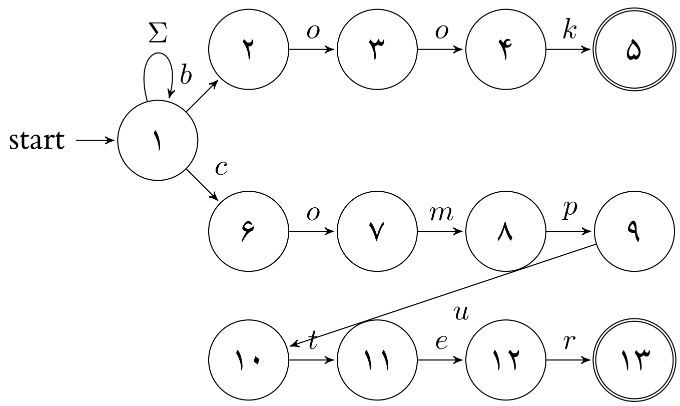
در این بخش کارکرد آتوماتون متناهی در جستجوی کلمه در یک متن توصیف می‌شود. به عنوان مثال بررسی وجود کلمات کلیدی "computer" و "book" در متن ورودی زیر بررسی می‌شود.

There are a number of reasons why computers won't replace books entirely. One reason is that books on paper are cheaper than computer. Books don't need a power source either.

ابتدا آتوماتون متناهی غیرقطعی M پذیرنده کلمات "computer" و "book" طراحی می‌شود (به عبارت دیگر M چنان به دست می‌آید که $L(M) = \{book, computer\}$). دیاگرام انتقال M در شکل ۱۱ آورده شده است.

با استفاده از روش توصیف شده در مثال ۱۴.۲، M تبدیل به آتوماتون متناهی قطعی $M_۱$ می‌شود. مجموعه حالت‌های $M_۱$، همان مجموعه توانی $\mathcal{P}(\{۱, ۲, \ldots, ۱۳\})$ است. محاسبه جدول انتقال $M_۱$، فقط با حالت‌های دسترس‌پذیر و لازم، با محاسبه مقادیر $\delta_۱(۱, \sigma) = \delta(۱, \sigma)$ شروع می‌شود. در جدول ۳ سطر اول نشانگر همین محاسبه است. حالت‌های جدید مورد نیاز $\{۱, ۶\}$ و $\{۱, ۲\}$ هستند که در جدول به صورت ۱, ۶ و ۱, ۲ نشان داده شده‌اند. با محاسبه مقادیر حالت‌های جدید سطرهای بعدی جدول ایجاد می‌شود. جدول ۳ توصیف تابع انتقال $M_۱$ است که در آن هر حالت یک زیر مجموعه از $\{۱, ۲, \ldots, ۱۳\}$، Σ مجموعه تمام نمادهای موجود در متن و $A = \{b, o, k, c, m, p, u, t, e, r\}$. حالت ۱, ۱۳ نشان دهنده مشاهده کلمه computer در متن است. همچنین حالت ۱, ۵ نشان‌دهنده مشاهده کلمه book در متن است. اکنون با استفاده از جدول انتقال آتوماتون قطعی $M_۱$ و جستجوی حرف به حرف متن ورودی، حالت ۱, ۱۳ دوبار و حالت ۱, ۵ سه بار در پردازش کامل متن ظاهر می‌شود که نشان دهنده دوبار حضور کلمه computer و سه بار حضور کلمه book در متن داده شده است.

از روش فوق برای جستجوی کلمات کلیدی دلخواه در هر متن استفاده می‌شود.



شکل ۱۱: دیاگرام انتقال M مربوط به بخش ۴

جدول ۳: جدول انتقال M_1 مربوط به شکل ۱۱

	$\Sigma \setminus A$	r	e	t	u	p	m	c	k	o	b
$\to$ ۱	۱	۱	۱	۱	۱	۱	۱	۱, ۶	۱	۱	۱, ۲
۱, ۲	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۳	۱, ۲
۱, ۶	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۷	۱, ۲
۱, ۳	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۴	۱, ۲
۱, ۷	۱	۱	۱	۱	۱	۱	۱, ۸	۱	۱	۱	۱, ۲
۱, ۴	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۵	۱	۱, ۲
۱, ۸	۱	۱	۱	۱	۱	۱, ۹	۱	۱	۱	۱	۱, ۲
۱, ۵*	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۲
۱, ۹	۱	۱	۱	۱	۱, ۱۰	۱	۱	۱	۱	۱	۱, ۲
۱, ۱۰	۱	۱	۱	۱, ۱۱	۱	۱	۱	۱	۱	۱	۱, ۲
۱, ۱۱	۱	۱	۱, ۱۲	۱	۱	۱	۱	۱	۱	۱	۱, ۲
۱, ۱۲	۱	۱, ۱۳	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۲
۱, ۱۳*	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱, ۲

۵. نتیجه‌گیری

استفاده از کامپیوتر در مسائل روزانه رو به افزایش است. بنابراین مطالعه مدل‌های مجرد از کامپیوتر و محاسبه لازم و ضروری است. در این مقاله، مفهوم آتوماتون متناهی قطعی و غیرقطعی و زبان‌های پذیرفته‌شده برای آن‌ها به صورت ریاضی توصیف شده است. با وجود تفاوت‌هایی در تابع انتقال آتوماتون متناهی قطعی و غیرقطعی، نشان داده شده است که مجموعه زبان‌های پذیرفته‌شده برای آتوماتای متناهی غیرقطعی، همان مجموعه زبان‌های پذیرفته‌شده برای آتوماتای متناهی قطعی است.

بسیاری از مسائل جهان واقعی، با استفاده از آتوماتون متناهی قطعی و غیرقطعی مدل‌سازی می‌شوند. در هر مسئله‌ای در ابتدا آتوماتون متناهی قطعی یا غیرقطعی متناظر با آن، به دست می‌آید. سپس آتوماتون فوق، به صورت الگوریتمی برای استفاده در نرم‌افزارهای مختلف استفاده می‌شود [۲۰]. در این مقاله، چگونگی طراحی دو آتوماتون متناهی برای دو مسئله مختلف توصیف شده است.

در مسئله اول، روند خرید اینترنتی، با استفاده از آتوماتون متناهی غیرقطعی مدل‌سازی شده است. با تدوین الگوریتم متناظر با آن در زبان‌های مختلف برنامه‌نویسی، نرم‌افزاری برای فروشگاه‌های مختلف به دست می‌آید که خرید اینترنتی را برای آن‌ها ممکن می‌سازد.

در مسئله دوم، آتوماتون متناهی قطعی مناسب برای جستجوی دو کلمه کلیدی در یک متن طراحی شده است. از این روش، می‌توان در جستجوی هر کلمه کلیدی، در هر متن استفاده کرد. از این نوع آتوماتون‌های متناهی در نرم افزارهای پویش متن‌های بزرگ مثل مجموعه صفحات وب برای یافتن وجود واژه‌ها و عبارات خاص استفاده می‌شود.

با وجود گستردگی کاربرد آتوماتای متناهی قطعی و غیرقطعی، حافظه کمکی در آن‌ها وجود ندارد. بنابراین در مسائلی که با شمارش و مقایسه مرتبط هستند، از آن‌ها استفاده نمی‌شود. به عنوان مثال زبان $L = \{a^n b^n : \ n \geq ۰\}$

[۲۰] در اینجا جزئیات مربوط به طراحی الگوریتم و برنامه نویسی بیان نمی‌شود. هدف اصلی ارائه جزئیات ریاضی برای آتوماتون متناهی قطعی و غیرقطعی است.

DOI: http://dx.doi.org/10.22108/msci.2020.120773.1354

توسط هیچ آتوماتون متناهی پذیرفته نمی‌شود. اگر به جای a از پرانتز چپ و به جای b از پرانتز راست استفاده شود، رشته‌های (()) و ((())) عضو L هستند؛ در حالی‌که رشته () عضو آن نیست. این نوع ساختارها در زبان‌های برنامه نویسی پیدا می‌شوند. بنابراین ابزاری قوی‌تر مورد نیاز است. آتوماتای پشته‌ای، توسیعی از آتوماتای متناهی قطعی و غیرقطعی است که در آن یک حافظه جانبی به نام پشته[۲۱] موجود است. در کارهای آتی، کاربردهای آتوماتای پشته‌ای در مسائل روزمره بررسی می‌شود.

مراجع

[1] M. Crochemore and T. Lecroq, *Pattern matching and text-compression algorithms*, The computer science and engineering Handbook, A. B Tucker, Jr. ed., CRC Press, Boca Raton, 2003.

[2] K. Culik II and J. Kari, *Image compression using weighted finite automata*, Mathematical foundations of computer science, Lecture Notes in Comput. Sci., 711, Springer, Berlin, (1993) 392–402.

[3] D. Ding-Zhu and K. Ker-I., *Problem Solving in Automata, Languages and Complexity*, 1st edition, New York, Wiley-Interscience, 2001.

[4] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, 1979.

[5] D. A. Huffman, The synthesis of sequential switching circuits, *J. Franklin Inst.*, 257 (1954) 275–303.

[6] P. Linz, *Introduction to Formal Languages and Automata*, 5th edition, Canada, Jones Bartlett Learning, 2011.

[7] J. C. Martin, *Introduction to Languages and The Theory of Computation*, 4th edition, New York, McGraw-Hill Education, 2010.

[8] G. Navarro and R. Baeza- Yates, Improving an Algorithm for Approximate Pattern Matching, *Algorithmica*, 30 (2001) 473–502.

[9] M. O. Rabin and D. Scott, Finite automata and their decision problems, *IBM J. Res. Develop*, 3 (1959) 114–125.

[10] M. Sipser, *Introduction to the Theory of Computation*, 3rd edition, United States, Cengage Learning, 2012.

سمیه تاری

گروه ریاضی– دانشکده علوم پایه– دانشگاه شهید مدنی آذربایجان– تبریز – ایران

s_tari @azaruniv.ac.ir

سمیه تاری کارشناسی خود را رشته ریاضی محض از دانشگاه تربیت معلم آذربایجان و کارشناسی ارشد و دکتری خود را در رشته ریاضی محض گرایش منطق ریاضی از دانشگاه تبریز اخذ کرده است. در حال حاضر استادیار دانشگاه شهید مدنی آذربایجان می‌باشد.

[۲۱] Stack